



UNITED STATES PATENT AND TRADEMARK OFFICE

A

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/049,211	02/05/2002	Siani Lynne Pearson	B-4487PCT 619499 -6	8087
36716	7590	10/18/2005		
LADAS & PARRY 5670 WILSHIRE BOULEVARD, SUITE 2100 LOS ANGELES, CA 90036-5679			EXAMINER MCKAY, KERRY A	
			ART UNIT 2131	PAPER NUMBER

DATE MAILED: 10/18/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No. 10/049,211	Applicant(s) PEARSON ET AL.	
	Examiner Kerry McKay	Art Unit 2131	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 05 February 2002.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-11, 19-22, and 26-47 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-11, 19-22 and 26-47 is/are rejected.
- 7) ☒ Claim(s) 1-2 is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☒ None of:
1. ☒ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. _____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| Paper No(s)/Mail Date <u>2/5/02-11/24/03</u> . | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. This is a non-final office action in response to the preliminary amendments filed on 02/05/2002. Claims 1-11, 19-22, and 26-47 are pending in this office action. Claims 12-18 and 23-25 have been cancelled.

Priority

2. Acknowledgment is made of applicant's claim for foreign priority based on an application filed in Europe on 8/13/1999. It is noted, however, that applicant has not filed a certified copy of the 99306415.3 application as required by 35 U.S.C. 119(b).

Claim Objections

3. Claims 1-2 are objected to because of the following informalities: awkward phraseology in the last limitation ("wherein the computer platform is programmed ...") of claim 1 suggests a method. Claim 2 builds upon this limitation with further method steps. Applicant is reminded that an apparatus and the method steps of using the apparatus is indefinite under U.S.C. 112 second paragraph. Further, such claims may be rejected under 101 as overlapping two statutory classes. Examiner recommends rewording the claim with means plus function in order to make clear that the steps described are part of the apparatus.

Claim Rejections - 35 USC § 112

The following is a quotation of the second paragraph of 35 U.S.C. 112:

Art Unit: 2131

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

4. Claim 28 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Claim 28 is a method claim which recites the structure of an apparatus, beginning at the transitional phrase "having" on the third line of the claim. Method steps are described later in the claim following the transitional phrase "comprising". For the purposes of examination, Examiner will treat all structural recitals as part of the preamble.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

5. Claim 29 is rejected under 35 U.S.C. 102(e) as being anticipated by England et al., U.S. Patent 6,820,063. Examiner notes that corresponding prior art terms are located beside the claim language in bracketed form.

Art Unit: 2131

6. Regarding claim 29, England et al. disclose a computer platform having:

a trusted module (DRMOS) which is resistant to internal tampering and which stores a third party's public key certificate (column 14, lines 19-21);

means for storing license-related code comprising, for at least one group of data, a software executor which specifies the respective group of data and which is operable to act as an interface to that group of data (column 19, lines 23-25, where "processing can be done on the content" denotes the software executor (application) is operable to act as an interface), the license-related code further comprising at least one of:

a secure executor for checking whether the platform or a user thereof is licensed to use particular data and for providing an interface for using the data (column 10, lines 26-36, and column 19, line 66 – column 20, line 2);

means for storing a hashed version of the license-related code signed with the third party's private key (figure 1B, items 142, 184, and 186);

wherein the computer platform is programmed so that, upon booting of the platform:

the license-related code is integrity checked with reference to the signed version and the public key certificate (figure 3 and column 11, lines 37-46); and

if the integrity check fails, the license-related code is prevented from being loaded (figure 3, and column 11, lines 37-46, where it is not loaded when the component is not necessary); and

wherein the means storing the license-related code is provided, at least in part, by the trusted module (DRMOS)(column 10, lines 26-36, where the license-related code is part of the trusted module).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

7. Claims 30, 31, 33, 35, 37, 38, and 40 are rejected under 35 U.S.C. 103(a) as being unpatentable over England et al., U.S. Patent 6,820,063.

8. Regarding per claim 30, England et al. teach the computer platform of claim 29.

The computer platform of England et al. further teaches:

the software executor (or at least one of the software executors) is operable to request the trusted module (DRMOS) to install particular data (England et al., column 10, lines 26-36, where each application is a software executor);

in response to such a request, the secure loader within the trusted module is operable to license-check whether the platform of a user thereof is licensed to install that particular data and/or to check the integrity of that data and to respond to the operating system with result of the check (column 10, lines 26-36); and

In dependence upon the response, the operating system is operable to install or not to install the particular data. An installer is an application that stores application data in the

operating system and on disk. It would have been obvious to one of ordinary skill in the art that the applications in the computer platform of England et al. include installers, and therefore, the applications are able to license-check the installations rights prior to allowing the installation by the operating system.

9. As per claim 31, the computer platform of England et al. teaches the computer platform of claim 29. The computer platform of England et al. further teaches that the operating system is programmed to install the particular data only in response to the trusted module (any code that wants to access the data, including the operating system, will be denied access to the particular data (content) if not permitted by the license (column 10, lines 26-36).

10. As per claim 33, the computer platform of England et al. teaches the computer platform of claim 29. The computer platform of England et al. further teaches that if the check succeeds, the trusted module is operable to generate a log for auditing the particular data (column 19, lines 25-28, where a log must be kept to determine how many times the content has been accessed).

11. Regarding claim 35, the computer platform of claim England et al. teaches the computer platform of claim 29. The computer platform of England et al. further teaches the particular data installed into the trusted module (DRMOS)(where the trusted module is the operating system, and data is typically installed into the operating system).

12. As per claim 37, the computer platform of England et al. teaches the computer platform of claim 29, where it is suggested that the secure executor contains at least one licensing model, as it can check a license to verify access rights (column 19, line 66 – column 20, line 2). Further, it is obvious that the operating system is operable to request the secure executor that a particular data be used (column 19, line 66 – column 20, line 2, where the application does so upon the operating system's request to run the program); and in response to such a request, the secure executable is operable to perform a license check using one of the license models (column 19, line 66 – column 20, line 2), and upon successful license-check, to request the operating system use that data.

13. As per claim 38, the computer platform of England et al. teaches the computer platform of claim 37. The computer platform of England et al. further teaches that the operating system is programmed to use the particular data only in response to the trusted module (any code that wants to access the data, including the operating system, will be denied access to the particular data (content) if not permitted by the license (column 10, lines 26-36).

14. As per claim 40, the computer platform of England et al. teaches the computer platform of claim 37. The computer platform of England et al. further teaches that the trusted module is operable to log the request to the operating system to use the data

Art Unit: 2131

(column 19, lines 25-28, where a log must be kept to determine how many times the content has been accessed).

15. Claims 1, 3-6, 8, 9, 21, 22, and 26 are rejected under 35 U.S.C. 103(a) as being unpatentable over England et al., U.S. Patent 6,820,063, in view of Graunke et al., U.S. Patent 5,991,399.

16. Regarding claim 1, England et al. disclose a computer platform having:
means storing license-related code (application) comprising at least one of:
a secure executor for checking whether the platform is licensed to use particular data and for providing an interface for using the data (column 10, lines 26-36, and column 19, line 66 – column 20, line 2); and
means storing a hashed version of the license-related code signed with the third party's private key (figure 1B, items 142, 184, and 186);
wherein the computer platform is programmed so that, upon booting of the platform:
the license-related code is integrity checked with reference to the signed version and the public key certificate (figure 3 and column 11, lines 37-46); and
if the integrity check fails, the license-related code is prevented from being loaded (figure 3, and column 11, lines 37-46, where it is not loaded when the component is not necessary). The computer platform of England et al. does not teach a trusted module (tamper resistant key module) which is resistant to internal tampering and which stores a third party's public key certificate.

Graunke et al. teach a trusted module (tamper resistant key module) which is resistant to internal tampering (column 7, line 31) and which stores a third party's public key certificate (column 7, lines 31-36). Graunke et al. further provide the motivation that a key module verifies the authenticity of a software executor (storage device reader) and that access to the content is allowed and that making it tamper resistant ensures that an attacker will not be able to modify the integrity parameters or otherwise alter the key module (column 4, lines 48-50, lines 64-67, column 5, lines 1-2, lines 11-14). It would have been obvious to a person of ordinary skill in the art at the time of applicant's invention to use the tamper-resistant module (key module) of Graunke et al. with the platform of England et al. to ensure the authenticity of the executor and that access to the content is allowed.

17. As per claim 3, the computer platform of England et al. and Graunke et al. discloses the computer platform of claim 1. The computer platform of England et al. and Graunke et al. further discloses secure key-transfer code (England et al., column 20, lines 21-27, where the content is transferred in encrypted format) for enabling a license key to be transferred between the trusted module and a further trusted module of another computer platform (sublicensing rights) (England et al., column 19 lines 32-36).

18. As per claim 4, the computer platform of England et al. and Graunke et al. teaches the computer platform of claim 1. The computer platform of England et al. and Graunke et al. further teaches that license-related code includes a library of interface

subroutines which can be called in order to communicate with the trusted module (England et al., column 20, lines 3-4).

19. Regarding claim 5, the computer platform of England et al. and Graunke et al. teaches the computer platform of claim 1. The computer platform of England et al. and Graunke et al. further teaches a software executor (application) which specifies the respective group of data (interpreted by Examiner as a type of data, such as those denoted by specific extensions) and which is operable to act as an interface to that group of data (England et al., column 19, lines 23-25, where “processing can be done on the content” denotes the software executor (application) is operable to act as an interface).

20. Regarding claim 6, the computer platform of England et al. and Graunke et al. teaches the computer platform of claim 1. The computer platform of England et al. and Graunke et al. further teaches means storing the license-related code are provided, at least in part, by the trusted module (key module) (Graunke et al., column 5, lines 18-22, and column 8, line 41, where the code used to verify integrity is stored in the key module).

21. Regarding claim 8, the computer platform of England et al. and Graunke et al. teaches the computer platform of claim 1.

Art Unit: 2131

the operating system is operable to request the secure loader (application) to license-check whether the platform or a user thereof is licensed to install that particular data (England et al., column 19, line 66 – column 20, line 2, where the application does so upon the operating system's request to run the program);

in response to such a request, the secure loader is operable to perform such a check and respond to the operating system with the result of the check (England et al., column 19, line 66 – column 20, line 2); and

in dependence upon the response, the operating system is operable to install or not to install the particular data. An installer is an application that stores application data in the operating system and on disk. It would have been obvious to one of ordinary skill in the art that the applications in the computer platform of England et al. and Graunke et al. include installers, and therefore, the applications are able to license-check the installations rights prior to allowing the installation by the operating system.

22. As per claim 9, the computer platform of England et al. teaches the computer platform of claim 8. It is not explicitly stated that the operating system is programmed to install the particular data only in response to the secure loader. However, it would have been obvious to one of ordinary skill in the art at the time of applicant's invention that any code that wants to access the data, including the operating system, will be denied access to the particular data (content) if not permitted by the license (England et al., column 10, lines 26-36).

23. As per claim 21, the computer platform of England et al. and Graunke et al. teaches the computer platform of claim 1. It is suggested that the secure executor contains at least one licensing model, as it can check a license to verify access rights (England et al., column 19, line 66 – column 20, line 2). Further, it is obvious that the operating system is operable to request the secure executor that a particular data be used (England et al., column 19, line 66 – column 20, line 2, where the application does so upon the operating system's request to run the program); and in response to such a request, the secure executable is operable to perform a license check using one of the license models (England et al., column 19, line 66 – column 20, line 2), and upon successful license-check, to request the operating system use that data.

24. As per claim 22, the computer platform of England et al. and Graunke et al. teaches the computer platform of claim 21. While it is not explicitly stated that the operating system is programmed to use the particular data only in response to the secure loader, it would have been obvious to one of ordinary skill in the art at the time of applicant's invention that any code that wants to access the data, including the operating system, will be denied access to the particular data (content) if not permitted by the license (England et al., column 10, lines 26-36).

25. Regarding claim 26, the computer platform of England et al. and Graunke et al. teaches the computer platform of claim 21. The computer platform of England et al. and Graunke et al. further teaches logging the request to the operating system to use the

data (England et al., column 19, lines 25-28, where each time the data is accessed, a counter increments). If this log (counter) were altered, a user could bypass restrictions on the number of uses by setting the counter back. This can be prevented by storing the log (counter) in a secure and trusted place, such as the trusted module. It would have been obvious for a person of ordinary skill in the art at the time of applicant's invention to have been motivated to perform this operation with the trusted module to prevent the modification of the log (counter) and ensure that the limit on the number of times a particular content is accessed is enforced.

26. Claims 2, 10, 11, 19, 20, 42, 43, 45, and 46 are rejected under 35 U.S.C. 103(a) as being unpatentable over England et al., U.S. Patent 6,820,063, and Graunke et al., U.S. Patent 5,991,399, as applied to claim 1 above, and further in view of Collins et al., U.S. Patent 6,378,072.

27. Regarding claim 2, the computer platform of England et al. and Graunke et al. teaches the computer platform of claim 1. The computer platform of England et al. and Graunke et al. does not teach the generation and comparison of first and second hashes.

Collins et al. teach reading and hashing the license-related code (program file) to produce a first hash; reading and decrypting the signed version using the public key certificate to produce a second hash; and comparing the first and second hashes (column 10, lines 8-16). Collins et al. further provide the motivation that an area of

concern is the secure loading storing of application programs, because if a program can be altered or substituted, then security may be breeched (column 2, lines 25-28). It would have been obvious to a person of ordinary skill in the art at the time of applicant's invention to use the hash comparison of Collins et al. with the computer platform of England et al. and Graunke et al. to verify that the license-related code (application) has not been altered or substituted.

28. Regarding claim 10, the computer platform of England et al. and Graunke et al. teaches the computer platform of claim 9. The computer platform of England et al. and Graunke et al. further teaches:

the trusted module stores a public key certificate for a party associated with the particular data to be installed (Graunke et al., column 7, lines 31-36);
the operating system is operable to include, in the request to check, the particular data together with a hashed version thereof signed with a private key of the associate party (England et al., column 7, lines 30-35). The computer platform of England et al. and Graunke et al. does not teach the secure loader being operable to hash the particular data included in the request to produce a third hash, decrypt the signed version in the request using the public key certificate for the associate party to produce a fourth hash, and to generate the response in dependence upon whether or not the third and fourth hashes match.

Collins et al. teach creating a hash, decrypting a digital signature with a public key to produce another hash, and determining if the two hashes match (column 10,

lines 8-16). Collins et al. further provide the motivation that an area of concern is the secure loading storing of application programs, because if a program can be altered or substituted, then security may be breeched (column 2, lines 25-28). It would have been obvious to a person of ordinary skill in the art at the time of applicant's invention to use the hash comparison of Collins et al. with the computer platform of England et al. and Graunke et al. to verify that the data has not been altered or substituted.

29. As per claim 11, the computer platform of England et al., Graunke et al., and Collins et al. teaches the computing platform of claim 10. The computer platform of England et al., Graunke et al., and Collins et al. further teaches that the license-related code includes, for at least one group of data, a (or a respective) software executor which specifies the respective group of data and which is operable to act as an interface to that group of data (England et al., column 19, lines 23-25, where "processing can be done on the content" denotes the software executor (application) is operable to act as an interface), and wherein the request to check includes the software executor for the particular data (column 10, lines 26-36).

30. As per claim 19, the computer platform of England et al. and Graunke et al. teaches the computer platform of claim 5. The computer platform of England et al. and Graunke et al. further teaches the software executor contains a licensing model for the respective data (England et al., column 10, lines 34-36, where it is necessary that a license model must be present in

order to determine the result of the license check, and the location of the license model is not specified and therefor may be located in the secure executor (DRMOS) or software executor (application));

the operating system is operable to request that software executor that its respective data be used (obvious, because software executors (applications) are initiated through an operating system);

in response to such a request, that software executor is operable to request the secure executor to license-check, using its licensing model, whether the platform or a user thereof is licensed to use that data (England et al., column 10, lines 26-36);

in response to such latter request, the secure executor is operable to perform the requested license-check (England et al., column 10, lines 26-36), to sign the result of the license check using a private key of the trusted module (the trusted module contains a private key, and the secure executor (DRMOS) is operable to request the trusted module to sign the result with that key, causing the result to be signed indirectly by the secure executor (DRMOS) with the private key of the trusted module), and to respond to that software executor with that signed result (England et al., column 10, lines 26-36, and figure 2, actions 7 and 9, where there is an intermediary component);

in response to such a response, that software executor is operable to check the integrity of the signed result using the public key of the trusted module (the processor has the capability to perform such operations (England et al., column 7, lines 30-35) and the software executor is therefore operable to check the signed result) and upon a successful integrity check of a successful license-check result, to request the operating

system to use that data (England et al., column 10, lines 26-36, where the operating system does not wait for a request, but the software executor is operable to do so).

The computer platform of England et al. and Graunke et al. does not teach the software executor containing a public key of the trusted module, the software executor using that public key to verify the integrity of the license-check result, or upon a successful integrity check of a successful license-check result, to request the operating system to use that data.

Collins et al. teach a stored public key and data signed with the corresponding private key (which produced a digital signature)(column 10, lines 10-13). Collins et al. further provide the motivation that this approach furnishes added confidence that the packet, and the content it contains, was generated from a source that was intended (column 10, lines 13-16). It would have been obvious to a person of ordinary skill in the art at the time of applicant's invention to sign the reply in the manner described by Collins et al. with the platform of England et al. and Graunke et al. to verify the integrity of the response. It would have further been obvious to store the public key of the trusted module in the software executor so that the software executor can verify the integrity of the response without the aid of other (possibly compromised) modules.

31. As per claim 20, the computer platform of England et al. and Graunke et al. teaches the computer platform of claim 5. The computer platform of England et al. and Graunke et al. further teaches

Art Unit: 2131

the software executor (or at least one of the software executors) contains a licensing model for the respective data (England et al., column 10, lines 34-36, where it is necessary that a license model must be present in order to determine the result of the license check, and the location of the license model is not specified and therefor may be located in the secure executor (DRMOS) or software executor (application);

the operating system is operable to request the secure executor that a particular data be used (obvious, because software executors (applications) are initiated through an operating system);

in response to such a request, the secure executor is operable to send to the respective software executor a request, signed using a private key of the trusted module, for a licensing model for the particular data (England et al., column 10, lines 26-36 and column 19, line 66 - column 20, line 2, where the secure executor and software executor can communicate with each other and are therefor operable to send requests between them);

in response to such latter request, that software executor is operable:

to check the integrity of the request using the public key of the trusted module (the processor has the capability to perform such operations (England et al., column 7, lines 30-35) and the software executor is therefore operable to check the integrity of the signed request); and upon a successful integrity check, to send the licensing model to the secure executor (England et al., column 10, lines 26-36, and column 19, line 66 - column 20, line 2, where the secure executor and software executor can communicate

with each other and are therefor operable to transfer data, such as a licensing model, between them); and
to perform a license-check using that licensing model (England et al., column 10, lines 26-36); and
upon a successful license-check, to request the operating system to use that data (England et al., column 10, lines 26-36, where the operating system does not wait for a request, but the software executor is operable to do so).
The computer platform of England et al. and Graunke et al. does not teach the software executor containing a public key of the trusted module.

Collins et al. teach a stored public key and data signed with the corresponding private key (which produced a digital signature)(column 10, lines 10-13). Collins et al. further provide the motivation that this approach furnishes added confidence that the packet, and the content it contains, was generated from a source that was intended (column 10, lines 13-16). It would have been obvious to a person of ordinary skill in the art at the time of applicant's invention to sign the request in the manner described by Collins et al. with the platform of England et al. and Graunke et al. to verify the integrity of the request. It would have further been obvious to store the public key of the trusted module in the software executor so that the software executor can verify the integrity of the response without the aid of other (possibly compromised) modules.

32. As per claim 42, the computer platform of England et al., Graunke et al., and Collins et al. teaches the computer platform of claim 19. While it is not explicitly stated

that the operating system is programmed to use the particular data only in response to the secure loader, it would have been obvious to one of ordinary skill in the art at the time of applicant's invention that any code that wants to access the data, including the operating system, will be denied access to the particular data (content) if not permitted by the license (England et al., column 10, lines 26-36).

33. Regarding claim 43, the computer platform of England et al., Graunke et al., and Collins et al. teaches the computer platform of claim 19. The computer platform of England et al., Graunke et al., and Collins et al. further teaches logging the request to the operating system to use the data (England et al., column 19, lines 25-28, where each time the data is accessed, a counter increments). If this log (counter) were altered, a user could bypass restrictions on the number of uses by setting the counter back. This can be prevented by storing the log (counter) in a secure and trusted place, such as the trusted module. It would have been obvious for a person of ordinary skill in the art at the time of applicant's invention to have been motivated to perform this operation with the trusted module to prevent the modification of the log (counter) and ensure that the limit on the number of times a particular content is accessed is enforced.

34. As per claim 45, the computer platform of England et al., Graunke et al., and Collins et al. teaches the computer platform of claim 20. While it is not explicitly stated that the operating system is programmed to use the particular data only in response to the secure loader, it would have been obvious to one of ordinary skill in the art at the

time of applicant's invention that any code that wants to access the data, including the operating system, will be denied access to the particular data (content) if not permitted by the license (England et al., column 10, lines 26-36).

35. As per claim 46, the computer platform of England et al., Graunke et al., and Collins et al. teaches the computer platform of claim 20. The computer platform of England et al., Graunke et al., and Collins et al. further teaches logging the request to the operating system to use the data (England et al., column 19, lines 25-28, where each time the data is accessed, a counter increments). If this log (counter) were altered, a user could bypass restrictions on the number of uses by setting the counter back. This can be prevented by storing the log (counter) in a secure and trusted place, such as the trusted module. It would have been obvious for a person of ordinary skill in the art at the time of applicant's invention to have been motivated to perform this operation with the trusted module to prevent the modification of the log (counter) and ensure that the limit on the number of times a particular content is accessed is enforced.

36. Claim 28 is rejected under 35 U.S.C. 103(a) as being unpatentable over England et al., U.S. Patent 6,820,063, in view of Fuh et al., U.S. Patent 6,463,474.

37. As per claim 28, England et al. disclose a method comprising the steps of: setting up secure communication between the trusted modules (page 13, paragraph 0154); sending the license or the key therefor from the first trusted module to the

second trusted module using the secure communication (page 13, paragraphs 0149 and 0154, where the license is transmitted with the content in encrypted form). England et al. do not teach deleting the license or the key therefor from the first trusted module.

Fuh et al. teach a step of deleting ACLs that are not in use (column 14, lines 42-47). The method of England et al. utilizes licenses or ACLs (page 5, paragraph 0067). Fuh et al. further provide motivation that doing so saves memory (column 14, line 46). It would have been obvious to a person of ordinary skill in the art at the time of applicant's invention to use the ACL deletion step of Fuh et al. with the method England to save memory by not storing licenses (or ACLs) that are no longer needed.

38. Claim 27 is rejected under 35 U.S.C. 103(a) as being unpatentable over England et al., U.S. Patent 6,820,063, and Graunke et al., U.S. Patent 5,991,399, as applied to claim 21 above, and further in view of DeTreville, U.S. Patent 6,609,199.

39. As per claim 27, the computer platform of England et al. and Graunke et al. teaches the computer platform of claim 21. The computer platform of England et al. and Graunke et al. teaches the secure executor or software executor is operable to perform the license-check with reference to the user identity (user-specific key)(England et al., column 10, lines 26-36, and column 17, lines 60-64). The computer platform of England et al. and Graunke et al. does not teach a further, removable, trusted module containing a user identity, wherein the platform is operable to perform an authentication check between the first-mention trusted module and the removable trusted module.

DeTreville teaches a removable trusted module (IC device) containing a user identity (column 1, lines 48-49 and 56-58); wherein the platform is operable to perform an authentication check between the first-mentioned trusted module and the removable trusted module (figure 13 and column 4, lines 35-43). DeTreville further provides the motivation that by coupling an IC device to a public computer, a user is able to access their private information or perform other restricted actions (such as gain access to content for which they hold a license). It would have been obvious to a person of ordinary skill in the art at the time of applicant's invention to use the removable trusted module (IC device) of DeTreville in conjunction with the computer platform of England et al. and Graunke et al. to allow users to access content for which they have a valid license on more than one system.

40. Claim 44 is rejected under 35 U.S.C. 103(a) as being unpatentable over England et al., U.S. Patent 6,820,063, Graunke et al., U.S. Patent 5,991,399, and Collins et al., U.S. Patent 6,378,072, as applied to claim 19 above, and further in view of DeTreville, U.S. Patent 6,609,199.

41. As per claim 44, the computer platform of England et al., Graunke et al., and Collins et al. teaches the computer platform of claim 19. The computer platform of England et al., Graunke et al., and Collins et al. teaches the secure executor or software executor is operable to perform the license-check with reference to the user identity (user-specific key)(England et al., column 10, lines 26-36, and column 17, lines 60-64).

The computer platform of England et al. and Graunke et al. does not teach a further, removable, trusted module containing a user identity, wherein the platform is operable to perform an authentication check between the first-mentioned trusted module and the removable trusted module.

DeTreville teaches a removable trusted module (IC device) containing a user identity (column 1, lines 48-49 and 56-58); wherein the platform is operable to perform an authentication check between the first-mentioned trusted module and the removable trusted module (figure 13 and column 4, lines 35-43). DeTreville further provides the motivation that by coupling an IC device to a public computer, a user is able to access their private information or perform other restricted actions (such as gain access to content for which they hold a license). It would have been obvious to a person of ordinary skill in the art at the time of applicant's invention to use the removable trusted module (IC device) of DeTreville in conjunction with the computer platform of England et al., Graunke et al., and Collins et al. to allow users to access content for which they have a valid license on more than one system.

42. Claim 47 is rejected under 35 U.S.C. 103(a) as being unpatentable over England et al., U.S. Patent 6,820,063, Graunke et al., U.S. Patent 5,991,399, and Collins et al., U.S. Patent 6,378,072, as applied to claim 20 above, and further in view of DeTreville, U.S. Patent 6,609,199.

43. Regarding claim 47, the computer platform of England et al., Graunke et al., and Collins et al. teaches the computer platform of claim 20. The computer platform of England et al., Graunke et al., and Collins et al. teaches the secure executor or software executor is operable to perform the license-check with reference to the user identity (user-specific key)(England et al., column 10, lines 26-36, and column 17, lines 60-64). The computer platform of England et al. and Graunke et al. does not teach a further, removable, trusted module containing a user identity, wherein the platform is operable to perform an authentication check between the first-mention trusted module and the removable trusted module.

DeTreville teaches a removable trusted module (IC device) containing a user identity (column 1, lines 48-49 and 56-58); wherein the platform is operable to perform an authentication check between the first-mentioned trusted module and the removable trusted module (figure 13 and column 4, lines 35-43). DeTreville further provides the motivation that by coupling an IC device to a public computer, a user is able to access their private information or perform other restricted actions (such as gain access to content for which they hold a license). It would have been obvious to a person of ordinary skill in the art at the time of applicant's invention to use the removable trusted module (IC device) of DeTreville in conjunction with the computer platform of England et al., Graunke et al., and Collins et al. to allow users to access content for which they have a valid license on more than one system.

44. Claims 36 and 41 are rejected under 35 U.S.C. 103(a) as being unpatentable over England et al., U.S. Patent 6,820,063 in view of DeTreville, U.S. Patent 6,609,199.

45. Regarding claim 36, the computer platform of England et al. teaches the computer platform of claim 29. The computer platform of England et al. does not teach a further, removable, trusted module, wherein the platform is operable to perform an authentication check between the first-mentioned trusted module and the removable trusted module, and wherein upon installation, the particular data is installed into the further trusted module.

DeTreville teaches a removable trusted module (IC device) (column 1, lines 48-49 and 56-58); wherein the platform is operable to perform an authentication check between the first-mentioned trusted module and the removable trusted module (figure 13 and column 4, lines 35-43). Code updates (such as firmware) may be installed into the further trusted module (column 12, lines 23-25). DeTreville further provides the motivation that by coupling an IC device to a public computer, a user is able to access their private information or perform other restricted actions (such as gain access to content for which they hold a license). It would have been obvious to a person of ordinary skill in the art at the time of applicant's invention to use the removable trusted module (IC device) of DeTreville in conjunction with the computer platform of England et al., Graunke et al., and Collins et al. to allow users to access content for which they have a valid license on more than one system.

46. As per claim 41, the computer platform of England et al. teaches the computer platform of claim 37. The computer platform of England et al. further teaches the secure executor or software executor is operable to perform the license-check with reference to the user identity (user-specific key)(column 10, lines 26-36, and column 17, lines 60-64). The computer platform of England et al. does not teach a further, removable, trusted module containing a user identity, wherein the platform is operable to perform an authentication check between the first-mention trusted module and the removable trusted module.

DeTreville teaches a removable trusted module (IC device) containing a user identity (column 1, lines 48-49 and 56-58); wherein the platform is operable to perform an authentication check between the first-mentioned trusted module and the removable trusted module (figure 13 and column 4, lines 35-43). DeTreville further provides the motivation that by coupling an IC device to a public computer, a user is able to access their private information or perform other restricted actions (such as gain access to content for which they hold a license). It would have been obvious to a person of ordinary skill in the art at the time of applicant's invention to use the removable trusted module (IC device) of DeTreville in conjunction with the computer platform of England et al. to allow users to access content for which they have a valid license on more than one system.

47. Claim 7 is rejected under 35 U.S.C. 103(a) as being unpatentable over England et al., U.S. Patent 6,820,063, and Graunke et al., U.S. Patent 5,991,399, in view of Johri et al., U.S. Patent 4,918,653.

48. Regarding claim 7, the computer platform of England et al. and Graunke et al. teaches the computer platform of claim 1. The computer platform does not teach the trusted module and operating system of the platform having a dedicated communications path which is inaccessible to other parts of the computer platform.

Johri et al. teach a trusted path that provides a non-forgable and non-penetrable communication path between a user terminal and the trusted operating system (column 19, lines 54-58). Johri et al. further provide the motivation that this mechanism guarantees that the data typed by a user is protected from any intrusion by unauthorized programs. It would have been obvious to one of ordinary skill in the art at the time of applicant's invention to use the trusted path mechanism of Johri et al. to provide secure and trusted communication between the trusted module and operating system of the computer platform of England et al. and Graunke et al. to prevent any intrusions from unauthorized programs which could circumvent access restrictions.

49. Claims 32 and 39 are rejected under 35 U.S.C. 103(a) as being unpatentable over England et al., U.S. Patent 6,820,063, in view of Johri et al., U.S. Patent 4,918,653.

50. Regarding claim 32, the computer platform of England et al. teaches the computer platform of claim 30. The computer platform of England et al. does not teach a dedicated communications path between the trusted module and operating system which is inaccessible to other parts of the communication platform.

Johri et al. teach a trusted path that provides a non-forgeable and non-penetrable communication path between a user terminal and the trusted operating system (column 19, lines 54-58). Johri et al. further provide the motivation that this mechanism guarantees that the data typed by a user is protected from any intrusion by unauthorized programs. It would have been obvious to one of ordinary skill in the art at the time of applicant's invention to use the trusted path mechanism of Johri et al. to provide secure and trusted communication between the trusted module and operating system of the computer platform of England et al. and Graunke et al. to prevent any intrusions from unauthorized programs which could circumvent the access restrictions.

51. As per claim 39, the computer platform of England et al. teaches the computer platform of claim 37. The computer platform of England et al. does not teach a dedicated communications path between the trusted module and operating system which is inaccessible to other parts of the communication platform.

Johri et al. teach a trusted path that provides a non-forgeable and non-penetrable communication path between a user terminal and the trusted operating system (column 19, lines 54-58). Johri et al. further provide the motivation that this mechanism guarantees that the data typed by a user is protected from any intrusion by

unauthorized programs. It would have been obvious to one of ordinary skill in the art at the time of applicant's invention to use the trusted path mechanism of Johri et al. to provide secure and trusted communication between the trusted module and operating system of the computer platform of England et al. and Graunke et al. to prevent any intrusions from unauthorized programs which could circumvent the access restrictions.

52. Claim 34 is rejected under 35 U.S.C. 103(a) as being unpatentable over England et al., U.S. Patent 6,820,063, in view of Rosenthal, U.S. Patent 5,359,659.

53. The computer platform of England et al. teaches the computer platform of claim 29. The computer platform of England et al. does not teach the secure loader being operable to perform a virus check.

Rosenthal teaches security routines which serve to verify the integrity of a program at execution time (column 11, lines 49-53). These security routines include a virus check (column 12, lines 16-17). Rosenthal further provides the motivation that this method protects existing executables against possible corruption (column 1, lines 25-30), and undetected infected programs can have devastating effects (column 1, lines 18-22). It would have been obvious to one of ordinary skill in the art at the time of applicant's invention to use the virus checking mechanism of Rosenthal with the computer platform of England et al. to prevent viruses from harming the platform.

Conclusion

Art Unit: 2131

54. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

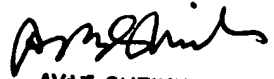
Lampson et al., U.S. Patent Application Publication 2003/0196085 A1: relates to the cited Patents of England et al. and DeTreville.

England et al., U.S. Patent 6,775,779: relates to the cited Patents of England et al. and DeTreville.

55. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Kerry McKay whose telephone number is (571)272-2651. The examiner can normally be reached on Monday-Friday, 8:00am-4:30pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Ayaz Sheikh can be reached on (571)272-3795. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).


AYAZ SHEIKH
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100